## LAB 10

**Objective**: Launch a DaemonSet, learn a few additional Pod specification parameters.

A DaemonSet is a **`v1beta1`** extension API resource. A DaemonSet will start one Pod on each node in your cluster (or a set of Pods selected by labels).

DaemonSets are used to run services daemons such as monitoring or network daemons. For example, in conjunction with **`kubeadm`**, you can use them to create a network overlay when building your cluster.

In your lab folder (in the *Course Resources*) you will find a basic DaemonSet manifest **`busy-daemon.yaml`**, it will run a sleeping **`busybox`** Pod with some special pod specification:

```
$ cat busy-daemon.yaml
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: busy-daemon
spec:
  template:
    metadata:
      labels:
        name: busy-daemon
    spec:
      hostNetwork: true
      hostPID: true
      containers:
```

 

```
    - name: busybox
      image: busybox
      command:
        - sleep
        - "3600"
      securityContext:
        privileged: true
```

Once you create it, you will see a Pod running in your minikube. Since minikube only has one node, you will not see other Pods. That Pod is special, as it has the network namespace of the host, the process namespace of the host and is a privileged container. This is a fairly *dangerous* container.

```
$ kubectl get daemonset
NAME          DESIRED   CURRENT   NODE-SELECTOR   AGE
busy-daemon   1         1         <none>          6s
$ kubectl get pods
NAME                          READY     STATUS              RESTARTS
AGE
busy-daemon-m3bcc             0/1       ContainerCreating   0
2s
```

Verify that it has the network configuration of the host:

```
$ kubectl exec -ti busy-daemon-m3bcc -- ifconfig
```

Verify that it has the process namespace of the host:

```
$ kubectl exec -ti busy-daemon-m3bcc -- ps -ef | grep docker
```

With such a Pod you can imagine running all your cluster daemons the same way that you now run distributed applications on Kubernetes.

 